Differentially Private Online Submodular Maximization

Sebastian Perez-Salazar
Industrial and Systems Engineering
Georgia Institute of Technology

Rachel Cummings
Industrial Engineering and Operations Research
Columbia University

Abstract

In this work we consider the problem of online submodular maximization under a cardinality constraint with differential privacy (DP). A stream of T submodular functions over a common finite ground set U arrives online, and at each time-step the decision maker must choose at most k elements of U before observing the function. The decision maker obtains a profit equal to the function evaluated on the chosen set and aims to learn a sequence of sets that achieves low expected regret.

In the full-information setting, we develop an (ε, δ)-DP algorithm with expected $(1 - 1/e)$-regret bound of $\mathcal{O}\left(\frac{k^2 \log |U| \sqrt{T \log k/\delta}}{\varepsilon}\right)$. This algorithm contains k ordered experts that learn the best marginal increments for each item over the whole time horizon while maintaining privacy of the functions. In the bandit setting, we provide an $(\varepsilon, \delta + O(e^{-T^{1/3}}))$-DP algorithm with expected $(1 - 1/e)$-regret bound of $\mathcal{O}\left(\frac{\sqrt{\log k/\delta}}{\varepsilon}(k(|U| \log |U|)^{1/3})^2 T^{2/3}\right)$. One challenge for privacy in this setting is that the payoff and feedback of expert i depends on the actions taken by her $i-1$ predecessors. This particular type of information leakage is not covered by post-processing, and new analysis is required. Our techniques for maintaining privacy with feedforward may be of independent interest.

1 INTRODUCTION

Ensuring users' privacy has become a critical task in online learning algorithms. As an illustrative example, sponsored search engines aim to maximize the probability that displayed ads or products are clicked by incoming customers, but prospective customers do not want their privacy infringed after clicking on a product. Users visiting online retailer web-pages such as Amazon, Walmart or Target leave behind an abundance of sensitive personal information that can be used to predict their behaviors or preferences, potentially leading to catastrophic results (Zhang et al., 2014)[1]. In this work, we introduce the first algorithms for privacy-preserving online monotone submodular maximization under a cardinality constraint.

A *submodular* set function $f : 2^U \to \mathbb{R}$ exhibits *diminishing returns*, meaning that adding an element x to a larger set B creates less additional value than adding x any subset of B. (See Definition 1 in Section 2 for a formal definition.) Submodular functions have found widespread application in economics, computer science and operations research (see, e.g., Bach et al. (2013) and Krause and Golovin (2014)), and have recently gained attention as a modeling tool for data summarization and ad display (Ahmed et al., 2012; Streeter et al., 2009; Badanidiyuru et al., 2014). We additionally consider *monotone* submodular functions, where adding elements to a set can only increase the value of f. Since unconstrained monotone submodular maximization is trivial—$f(S)$ can be maximized by choosing the entire universe $S = U$—we consider *cardinality constrained* maximization, where the decision-maker solves: $\max_{S \subseteq U} f(S)$ s.t. $|S| \leq k$.

In the online learning setting, at each time-step t a learner must choose a set $S_t \subseteq U$ of size at most k and receives payoff $f_t(S_t)$ for a monotone submodular function f_t. Importantly, the learner does not know f_t before she chooses S_t, but this set can be chosen based on previous functions $f_1, \ldots, f_{t-1}$. Two types of infor-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

[1]See also `https://www.nytimes.com/2012/02/19/magazine/shopping-habits.html`

mational feedback are commonly studied in the online learning literature. In the *full-information* setting, the learner gets full oracle access to the function f_t after choosing S_t, and thus is able to incorporate the entirety of previous functions into her future decisions. In the *bandit* setting, the learner only observes her own payoff $f_t(S_t)$ as feedback.

Performance of an online learner is typically measured by the *regret*, which is the difference between the best fixed decision in hindsight and the cumulative payoff obtained by the learner (Zinkevich, 2003; Hazan et al., 2016; Shalev-Shwartz et al., 2012). More precisely, the regret of a learner after T rounds is: $\max_{|S|\leq k}\sum_{t=1}^{T} f_t(S) - \sum_{t=1}^{T} f_t(S_t)$. The aim often is to design algorithms with *sublinear* regret, i.e., $o(T)$, so that the average payoff over time of the algorithm is comparable with the best average fixed profit in hindsight. Offline monotone submodular maximization under a cardinality constraint is NP-hard to approximate with a factor better than $(1-1/e)$ (Feige, 1998; Mirrokni et al., 2008), so we instead measure the quality of our algorithms using the more restrictive notion of $(1-1/e)$-regret (Streeter and Golovin, 2009; Streeter et al., 2009):

$$\mathcal{R}_T = \left(1-\frac{1}{e}\right)\max_{|S|\leq k}\sum_{t=1}^{T} f_t(S) - \sum_{t=1}^{T} f_t(S_t). \qquad (1)$$

The privacy notion we consider in this work is *differential privacy* (Dwork et al., 2006), which enables accurate estimation of population-level statistics while ensuring little can be learned about the individuals in the database. Informally, a randomized algorithm is said to be differentially private if changing a single entry in the input database results in only a small distributional change in the outputs. (See Definition 2 in Section 2 for a formal definition.) This means that an adversary cannot information-theoretically infer whether or not a single individual participated in the database. Differentially private algorithms have been deployed by major organizations including Apple, Google, Microsoft, Uber, and the U.S. Census Bureau, and are seen as the gold standard in privacy-preserving data analysis. In this work, the input database to our learning algorithm consists of a stream of functions $F = \{f_1, \ldots, f_T\}$, and each individual's data corresponds to a function f_t. Our privacy guarantees ensure that the stream of chosen sets $S_1, \ldots, S_T$ are differentially private with respect to this database of functions,

In both the full-information and bandit settings, we present differentially private online learning algorithms that achieve sublinear expected $(1-1/e)$-regret.

Motivating Example. While there are countless examples of practical online submodular maximization problems using sensitive data, we offer this motivating example for concreteness. Consider an online product display model where a website has k display slots and wants to maximize the probability of any displayed product being clicked. Each customer t has a (privately known) probability p_a^t of clicking a display for product $a \in U$, independently of the other products displayed. Let $f_t(S)$ denote the probability that customer t clicks on any product in a display set S. We can write this function in closed form as $f_t(S) = 1 - \prod_{a\in S}(1-p_a^t)$. Note that this function is submodular because adding products to the set S exhibits diminishing returns in total click probability. Each customer's click-probabilities $\{p_a^t\}_{a\in U}$ contain sensitive information about his preferences or habits, and require formal privacy protections.

1.1 Our Results

Our main results are differentially private algorithms for online submodular maximization under a cardinality constraint. We provide algorithms that achieve sublinear expected $(1-1/e)$-regret in both the full-information and bandit settings.

Our algorithms are based on the approach of Streeter and Golovin (2009), who designed (non-private) online algorithms with low expected $(1-1/e)$-regret for submodular maximization. We adapt and extend their techniques to additionally satisfy differential privacy. Following the spirit of Streeter and Golovin (2009), our algorithms have k ordered online learning algorithms, or *experts*, that together pick k items at every time-step and learn from their decisions over time. Roughly speaking, expert i learns how to choose an item that complements the decisions of the previous $i-1$ experts. The expected $(1-1/e)$-regret can be bounded by the regret of these k experts, so to show a low $(1-1/e)$-regret algorithm that preserves privacy, we simply need to find no-regret experts that together preserve privacy. Ideally, we would like each expert to be differentially private so that simple composition and post-processing arguments would yield overall privacy guarantees. Unfortunately this is not possible for $k > 1$ because the choices of all previous experts alter the distribution of payoffs for expert i.

Specifically, the i-th expert non-privately queries the function (i.e., accesses the database) at $|U|$ points that depend on the action of the previous experts. A naive solution is to allow each expert to query the function at any of its $2^{|U|}$ values, and then privacy would be satisfied by post-processing on the differentially private outputs of previous experts. However, this larger domain size requires large quantities of noise that would harm the experts' no-regret guarantees. Effectively, this decouples the advice of the k experts, so that experts are not learning from each other. This naturally helps privacy but harms learning. Instead, we restrict each expert to a domain of size $|U|$ that is defined by the actions of previous experts. This ensures no-regret learning, but post-processing no longer ensures privacy. We overcome this challenge by showing that *together* the experts are differentially private and sufficiently low quantities of noise are needed.

Theorem 1 below is an informal version of our main results in the full-information setting (Theorems 5 and 6 in Section 3).

Theorem 1 (Informal)**.** *In the full-information setting, Algorithm 2 for online monotone k-cardinality-constrained submodular maximization is (ε,δ)-differentially private and guarantees*

$$\mathbb{E}\left[\mathcal{R}_T\right] = \mathcal{O}\left(\frac{k^2 \log |U| \sqrt{T \log(k/\delta)}}{\varepsilon}\right).$$

In the bandit setting, each expert only receives its own payoff as feedback, and does not have oracle access to the entire function. For this setting, we modify the full-information algorithm by using a biased estimator of the marginal increments for other actions.

The algorithm also requires additional privacy considerations. The non-private approach of Streeter and Golovin (2009) randomly decides in each round whether to explore or exploit. In exploit rounds, the experts sample a new set but play the current-optimal action, providing both learning and exploitation. Directly privatizing this algorithm incurs additional privacy loss from the exploit rounds, which leads to a weak bound of $\mathcal{O}(T^{3/4})$ for the expected $(1-1/e)$-regret, far from the best known $\mathcal{O}(T^{2/3})$. Instead, we have the experts sample new sets only after an exploration round has occurred. The choice to explore is data-independent, so privacy is maintained by post-processing. If the exact number and timing of explore rounds are known in advance, this results in an (ε,δ)-DP algorithm. However, this approach requires $\Omega(T^{2/3}+k|U|)$ space, which is not appealing in practical settings where T is substantially larger than U. Instead we allow explore-exploit decisions to be made online and obtain a high probability bound on the number of explore rounds based on the sampling parameter. At the expense of an exponentially small loss in the δ privacy parameter—resulting from the failure of the high probability bound—we obtain the asymptotically optimal $\mathcal{O}(T^{2/3})$ expected $(1-1/e)$-regret.

Theorem 2 is an informal version of our main results in the more challenging bandit feedback setting (Theorems 7 and 8 in Section 4).

Theorem 2 (Informal)**.** *In the bandit feedback setting, Algorithm 3 for online monotone k-cardinality-constrained submodular maximization is $(\varepsilon,\delta + e^{-8T^{1/3}})$-differentially private and guarantees*

$$\mathbb{E}\left[\mathcal{R}_T\right] = \mathcal{O}\left(\frac{\sqrt{\log k/\delta}}{\varepsilon}(k(|U|\log|U|)^{1/3})^2 T^{2/3}\right).$$

The best known non-private expected $(1-1/e)$-regret in the full-information setting is $\mathcal{O}\left(\sqrt{kT\log|U|}\right)$ and in the bandit setting is $\mathcal{O}\left(k(|U|\log|U|)^{1/3}T^{2/3}\right)$ (Streeter and Golovin, 2009). Comparing our expected $(1-1/e)$-regret bounds to these, we see that our bounds match asymptotically the best known bounds in T, and have slight gaps in terms of k and U. Typically, the dominating term is the time horizon T with $k \leq |U| \ll T$, so our results match the best expected $(1-1/e)$-regret asymptotically in T. At each time step $t=1,\ldots,T$, our algorithms have time complexity $\mathcal{O}(k|U|)$.

Additionally, we show that our algorithms can be extended to a continuous generalization of submodular functions, know as *DR-submodular* functions. We provide a differentially private online learning algorithm for DR-submodular maximization that achieves low expected regret. A brief overview of this extension is given in Section 5, with further details in the appendix.

1.2 Related Work

Online learning (Zinkevich, 2003; Cesa-Bianchi and Lugosi, 2006; Hazan et al., 2016; Shalev-Shwartz et al., 2012) has gained increasing attention for making decisions in dynamic environments when only partial information is available. Its applicability in ad placement (Chatterjee et al., 2003; Chapelle and Li, 2011; Tang et al., 2014) has made this model attractive from a practical viewpoint.

Submodular optimization has been widely studied, due to the large number of important submodular functions, such as the cut of a graph, entropy of a set of random variables, and the rank of a matroid, to name only a few. For more applications see (Schrijver, 2003; Williamson and Shmoys, 2011; Bach et al., 2013). While (unconstrained) submodular minimization can be solved with polynomial number of oracle calls (Schrijver, 2003; Bach et al., 2013), submodular maximization is known to be NP-hard for general submodular functions. Nemhauser and Wolsey (1978) showed that algorithms that evaluate submodular functions in a polynomial number of sets cannot guarantee factors better than $(1-1/e)$ of the optimal value, even for monotone functions under cardinality constraint. The greedy algorithm (Fisher et al., 1978) achieves this factor. For further results with more general constraints, we refer the reader to the survey (Krause and Golovin, 2014). In the online setting, Streeter and Golovin (2009) and Streeter et al. (2009) were the first to study online monotone submodular maximization, respectively with cardinality/knapsack constraints and partition matroid constraints. Recently, continuous submodularity, has gained attention in the optimization community Bian et al. (2017); Hassani et al. (2017); Niazadeh et al. (2018); Zhang et al. (2020). See Chen et al. (2018a,b) for online continuous submodular optimization.

Differential privacy (Dwork et al., 2006) has become the gold standard for individual privacy, and there as been a large literature developed of differentially private algorithms for a broad set of analysis tasks. See Dwork and Roth (2014) for a textbook treatment. Due to privacy concerns in practical applications of online learning, there has been growing interest in implementing well-known methods—such as experts algorithms and gradient optimization methods–in a differentially private way. See for instance (Jain et al., 2012; Thakurta and Smith, 2013).

Differential privacy and submodularity were first jointly considered in (Gupta et al., 2010). They studied the combinatorial public projects problem, where the objective function was a sum of monotone submodular functions, each representing an agent's private valuation function, and a decision-maker must maximize this objective subject to a cardinality constraint. The authors designed an $(\varepsilon, 0)$-DP algorithm using the Exponential Mechanism of (McSherry and Talwar, 2007) as a private subroutine, and achieved a $(1-1/e)$-approximation to the optimal non-private solution, plus an additional $\propto \varepsilon^{-1}$ term. Later, Mitrovic et al. (2017) extended these results to monotone submodular functions in the cardinality, matroid and p-system constraint cases. Their methods also used the Exponential Mechanism to ensure differential privacy. See also recent work by Rafiey and Yoshida (2020).

In the online learning framework, Cardoso and Cummings (2019) study online (unconstrained) differentially private submodular minimization. They use the Lovász extension of a set function as a convex proxy to apply known privacy tools that work in online convex optimization (Jain et al., 2012; Thakurta and Smith, 2013). Since submodular minimization and maximization are fundamentally different technical problems, the techniques of Cardoso and Cummings (2019) do not extend to our setting.

Fundamental to our analysis is the differentially private Exponential Mechanism of McSherry and Talwar (2007) and its inherent connection to multiplicative weights algorithms (Hazan et al., 2016; Shalev-Shwartz et al., 2012) to estimate probability distributions in the simplex while preserving privacy.

2 PRELIMINARIES

In this section we review definitions and properties of submodular functions and differential privacy.

Definition 1 (Submodularity). *A function $f : 2^U \to \mathbb{R}$ is submodular if it satisfies the following diminishing returns property: For all $A \subseteq B \subseteq U$ and $x \notin B$,*[2]

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B).$$

As is standard in the submodular maximization literature, we assume $f(\emptyset) = 0$. In our motivating example, this means that if no items are shown to the incoming customer, then the probability of selecting an item is 0. We let $\mathcal{F}$ denote the family of submodular functions with finite ground set U. For the sake of simplicity, we will additionally assume that all functions take value in the interval $[0, 1]$. This does not change our analysis as long as the functions take value in a bounded interval $[0, M]$. Indeed, by rescaling appropriately the learning rates in our algorithms (see below), we obtain the same privacy guarantees, and regret guarantees up to a factor of M—expected since functions take values in $[0, M]$. In this work, we additionally consider set functions f that are monotone or non-decreasing, i.e., $f(A) \leq f(B)$ for all $A \subseteq B$.

In the problem of online monotone submodular maximization under a cardinality constraint, a sequence of T monotone submodular functions $f_1, \ldots, f_T : 2^U \to [0, 1]$ arrive in an online fashion. At every time-step t, the decision maker $\mathcal{A}$ has to choose a subset $S_t \subseteq U$ of size at most k before observing f_t. This decision must be based solely on previous observations. The decision maker $\mathcal{A}$ receives a payoff $f_t(S_t)$ and her goal is to minimize the $(1 - 1/e)$-expected-regret $\mathbb{E}[\mathcal{R}_T]$, where $\mathcal{R}_T = \left(1 - \frac{1}{e}\right) \max_{|S| \leq k} \sum_{t=1}^{T} f_t(S) - \sum_{t=1}^{T} f_t(S_t)$ as defined in Equation (1), and the randomness is over the algorithm's choices.

A fundamental tool in our analysis is the Hedge algorithm (Algorithm 1) of Freund and Schapire (1997) which chooses an action from a set $[N] = \{1, \ldots, N\}$ based on past payoffs from each action. The algorithm takes as input a learning rate η and a stream of linear functions $g_1, \ldots, g_T : [N] \to [0, 1]$, where the payoff of playing action i at time t is $g_t(i)$.

In our setting, the learner must select a set of at most k items from the ground set U. The learner does this by

[2] Equivalently, f is submodular if $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$ for all $A, B \subseteq U$.

implementing k ordered copies of the Hedge algorithm, each of which choses one item, so the action space for each instantiation is the ground set: $N = U$. The i-th copy of Hedge learns the item with the best marginal gain given the decisions made by the previous $i-1$ Hedge algorithms.

Algorithm 1: HEDGE$(\eta, g_1, \dots, g_T)$

Initialize $w_1 = (1, \dots, 1) \in \mathbb{R}^N$
for $t = 1, \dots, T$ **do**
  Sample action $i_t \in [N]$ w.p. $x_t(i) = \frac{w_t(i)}{\sum_j w_t(j)}$
  Obtain payoff $g_t(i_t)$ and full access to g_t
  Update $w_{t+1}(i) = w_t(i) e^{\eta g_t(i)}$

The Hedge algorithm exhibits the following guarantee, which is useful for analyzing its regret, as well as the regret of our algorithms which instantiate Hedge.

Theorem 3 (Freund and Schapire (1997))**.** *For any $i \in [N]$, the distributions $\mathbf{x}_1, \dots, \mathbf{x}_T$ over $[N]$ constructed by Algorithm 1 satisfy*

$$\sum_{t=1}^{T} g_t(i) - \sum_{t=1}^{T} \mathbf{x}_t^\top g_t \le \eta \sum_{t=1}^{T} \mathbf{x}_t^\top g_t^2 + \frac{\log N}{\eta},$$

where g_t^2 is the vector g_t with each coordinate squared.

For the privacy considerations of this work, we view the input database as the ordered input sequence of submodular functions $F = \{f_1, \dots, f_T\}$ and the algorithm's output as the sequence of chosen sets $S_1, \dots, S_T$. We say that two sequences F, F' of functions are *neighboring* if $f_t \neq f_t'$ for at most one $t \in [T]$.

Definition 2 (Differential Privacy (Dwork et al., 2006))**.** *An online learning algorithm $\mathcal{A} : \mathcal{F}^T \to (2^U)^T$ is (ε, δ)-differentially private if for any neighboring function databases F, F', and any event $S \subseteq (2^U)^T$,*

$$\Pr(\mathcal{A}(F) \in S) \le e^{\varepsilon} \Pr(\mathcal{A}(F') \in S) + \delta.$$

Differential privacy is robust to post-processing, meaning that any function of a differentially private output maintains the same privacy guarantee.

Proposition 1 (Post-Processing (Dwork et al., 2006))**.** *Let $\mathcal{M} : \mathcal{F}^T \to \mathcal{R}$ be an (ε, δ)-DP algorithm and let $h : \mathcal{R} \to \mathcal{R}'$ be an arbitrary function. Then, $\mathcal{M}' \doteq h \circ \mathcal{M} : \mathcal{F}^T \to \mathcal{R}'$ is also (ε, δ)-DP.*

Differentially private algorithms also *compose*, and the privacy guarantees degrade gracefully as addition DP computations are performed. This enables modular algorithm design using simple differentially private building blocks. *Basic Composition* (Dwork et al., 2006) says that can simply add up the privacy parameters used in an algorithm's subroutines to get the overall privacy guarantee. The following Advanced Composition theorem provides even tighter bounds.

Theorem 4 (Advanced Composition (Dwork et al., 2010b))**.** *Let $\mathcal{M}_1, \dots, \mathcal{M}_k$ each be (ε, δ)-DP algorithms. Then, $\mathcal{M} = (\mathcal{M}_1, \dots, \mathcal{M}_k)$ is $(\varepsilon', k\delta + \delta')$-DP for $\varepsilon' = \sqrt{2k \log(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1)$ and any $\delta' \ge 0$.*

Our algorithms rely on the Exponential Mechanism (EM) introduced by McSherry and Talwar (2007). The EM takes in database F, a finite action set U, and a quality score $q : \mathcal{F}^T \times U \to \mathbb{R}$, where $q(F, i)$ assigns a numeric score to the quality of outputting i on input database F. The *sensitivity* of the quality score, denoted Δq, is the maximum change in the value of q across neighboring databases: $\Delta q = \max_{i \in U} \max_{F, F' \text{ neighbors}} |q(F, i) - q(F', i)|$. Given these inputs, the EM outputs $i \in U$ with probability proportional to $\exp(\varepsilon \frac{q(F,i)}{2\Delta q})$. The Exponential Mechanism is $(\varepsilon, 0)$-DP (McSherry and Talwar, 2007).

As noted by Jain et al. (2012) and Dwork et al. (2010a), the Hedge algorithm can be converted into a DP algorithm using advanced composition and EM.

Proposition 2. *If $\eta = \frac{\varepsilon}{\sqrt{32T \log 1/\delta}}$, Hedge (Algorithm 1) is (ε, δ)-DP.*

3 FULL INFORMATION SETTING

In this section, we introduce our first algorithm for online submodular maximization under cardinality constraint. It is both differentially private and achieves the best known expected $(1-1/e)$-regret in T. For cardinality k, the learner implements k ordered copies of the Hedge algorithm. Each copy is in charge of learning the marginal gain that complements the choices of the previous Hedge algorithms. At time-step t, each Hedge algorithm selects an element $a \in U$ and the learner gathers these choices to play the corresponding set. When she obtains oracle access to the submodular function, for each $i \in [k]$, she constructs a vector g_t^i with a-th coordinate given by the marginal gain of adding $a \in U$ to the choices made by the previous $i-1$ Hedge algorithms. Finally, she feeds back the vector g_t^i to Hedge algorithm i. A formal description of this procedure is presented in Algorithm 2.

To ensure differential privacy, it would be enough to show that each Hedge $\mathcal{E}_i$ is $(\varepsilon/k, \delta/k)$-DP. Indeed, if the sequence $(a_1^i, \dots, a_T^i)$ constructed by each Hedge algorithm i is $(\varepsilon/k, \delta/k)$-DP, then by Basic Composition and post-processing, the sequence $(S_1, \dots, S_T)$ is (ε, δ)-DP, where $S_t = \{a_t^i\}_{i=1}^k$. However, for $i \ge 2$, the output of expert $\mathcal{E}_i$ depends on the choices made by algorithms $\mathcal{E}_1, \dots, \mathcal{E}_{i-1}$. Moreover, algorithm $\mathcal{E}_i$ by itself is again accessing the database F, hence ruling out a post-processing argument. More specifically, $\mathcal{E}_i$ takes as input not just the private output of $\mathcal{E}_1, \dots, \mathcal{E}_{i-1}$,

---

Algorithm 2: FI-DP($F = \{f_t\}_{t=1}^T, k, \varepsilon, \delta$)

Initialize: Set $\eta = \frac{\varepsilon}{k\sqrt{32T\log(k/\delta)}}$

Instantiate k parallel copies $\mathcal{E}_1, \ldots, \mathcal{E}_k$ of Hedge algorithm with rate η.

for $t = 1, \ldots, T$ **do**

  For each $i = 1, \ldots, k$, sample a_t^i given by $\mathcal{E}_i$.

  Play $S_t = \cup_{i=1}^k \{a_t^i\}$.

  Obtain $f_t(S_t)$ and oracle access to f_t.

  For each $i = 1, \ldots, k$, define linear function $g_t^i : U \to [0, 1]$:

$$g_t^i(a) = f_t(S_t^{i-1} + a) - f_t(S_t^{i-1}), \quad \forall a \in U,$$

  where $S_t^i = \cup_{j=1}^i \{a_t^j\}$.

  Feed back each Hedge algorithm $\mathcal{E}_i$ with g_t^i

---

namely S_t^{i-1}, but also a function of this private output that also depends on the database, namely the vector g_t^i. This precludes using post-processing arguments to show privacy of $\mathcal{E}_i$. Despite this, we show that all experts together are (ε, δ)-DP even though individually we cannot ensure they preserve $(\varepsilon/k, \delta/k)$-DP.

It is worth noting that the Hedge algorithms $\mathcal{E}_1, \ldots, \mathcal{E}_k$ in Algorithm 2 can be replaced by any other no-regret DP method that selects items over U, and the same proof structure would follow—although the regret bound would depend on the choice of no-regret algorithm. For instance, if we utilize the private experts method of(Thakurta and Smith, 2013) instead of the Hedge algorithm, Algorithm 2 would be $(\varepsilon, 0)$-DP with a regret bound of $\mathcal{O}\left(k^2 \frac{\sqrt{|U| T \log^{2.5} T}}{\varepsilon}\right)$.

Theorem 5. *Algorithm 2 is (ε, δ)-differentially private.*

Theorem 6. *Algorithm 2 has $(1 - 1/e)$-expected-regret*

$$\mathbb{E}\left[\mathcal{R}_T\right] \leq \mathcal{O}\left(\frac{k^2 \log|U| \sqrt{T\log(k/\delta)}}{\varepsilon}\right).$$

Proof of Theorem 5 The output of Algorithm 2 is the stream of sets $(S_1, \ldots, S_T)$. Before showing that this output preserves privacy, we deal with a simpler case from which we can deduce an inductive argument.

Note that $\mathcal{E}_1(F)$ receives as feedback the functions $g_t^1 = (f_t(a))_{a \in U}$ at each time step. By Proposition 2, we have that $\mathcal{E}_1$ is $(\varepsilon/k, \delta/k)$-DP given that $\eta = \frac{\varepsilon}{k\sqrt{32T\log k/\delta}}$. On the other hand $\mathcal{E}_2(F)$ receives as feedback the functions $g_t^2 = (f_t(a_t^1 + a) - f_t(a_t^1))_{a \in U}$ at each time-step, where a_t^1 is computed by $\mathcal{E}_1(F)$. Therefore, the output of $\mathcal{E}_2$ depends uniquely on the choices of $\mathcal{E}_1$, hence, conditioning on these choices, $\mathcal{E}_2$ should also be $(\varepsilon/k, \delta/k)$-DP. We generalize and formalize this in the next few paragraphs.

Consider the following family of algorithms: For $a^1, \ldots, a^{i-1} \in U^T$ let $S^{i-1} = \{a^{i-1}, \ldots, a^1\}$. For $t = 1, \ldots, T$, let $\mathcal{M}_t^{S^{i-1}} : \mathcal{F}^T \to \Delta(U)$ be the EM that outputs $a \in U$ with probability proportional to $e^{\eta \sum_{\tau < t} f_\tau(S_\tau^{i-1} \cup \{a\}) - f_\tau(S_\tau^{i-1})}$. Each of these mechanisms is 2η-DP by Proposition 2. Therefore, by Advanced Composition and our choice of η, $\mathcal{M}^{S^{i-1}} := (\mathcal{M}_1^{S^{i-1}}, \ldots, \mathcal{M}_T^{S^{i-1}})$ is $(\varepsilon/k, \delta/k)$-DP. Note that for $S \subseteq U^T$ we have

$$\begin{aligned}&\Pr(\mathcal{E}_i(F) \in S \mid (\mathcal{E}_{i-1}, \ldots, \mathcal{E}_1)(F) = S^{i-1}) \\ &= \Pr(\mathcal{M}^{S^{i-1}}(F) \in S)\end{aligned}$$

and the latter expression describes the output of an $(\varepsilon/k, \delta/k)$-DP algorithm. This formalizes the idea that $\mathcal{E}_2$ is $(\varepsilon/k, \delta/k)$-DP if the choices of $\mathcal{E}_1$ are fixed. We utilize this idea to show that *together* $(\mathcal{E}_k, \ldots, \mathcal{E}_1)$ are (ε, δ)-DP. This is formally presented in Lemma 1. The proof of this result (formally given in Appendix A.1) is an inductive argument that takes advantage of the DP guarantee of the mechanisms $\mathcal{M}^{S^{i-1}}$.

Lemma 1. *For any $i \in [k]$, the function $(\mathcal{E}_i, \mathcal{E}_{i-1}, \ldots, \mathcal{E}_1) : \mathcal{F}^T \to U^T \times \cdots \times U^T$ which is the composition of the first i Hedge algorithms is $(i\varepsilon/k, i\delta/k)$-DP.*

Lemma 1 with $i = k$ and post-processing ensures that Algorithm 2 is (ε, δ)-DP. □

Proof of Theorem 6 The key idea is to bound the $(1 - 1/e)$-regret of Algorithm 2 by the regret incurred by the k Hedge algorithms $\mathcal{E}_1, \ldots, \mathcal{E}_k$. We formalize this in Proposition 3 below. With this bound, we can utilize the regret bound of the Hedge algorithm and conclude the proof. The regret incurred by $\mathcal{E}_i$ is

$$r_i = \max_{a \in U} \sum_{t=1}^T g_t^i(a) - \sum_{t=1}^T g_t^i(a_t).$$

where $g_t^i = (f_t(S_t^{i-1} \cup \{a\}) - f_t(S_t^{i-1}))_{a \in U}$.

Proposition 3. *The $(1 - 1/e)$-regret of Algorithm 2 is bounded by the expected regret of $\mathcal{E}_1, \ldots, \mathcal{E}_k$.*

While a full proof of Proposition 3 is deferred to in Appendix A.2, we describe the key idea here. To bound the $(1 - 1/e)$-regret, we rewrite the regret r_i via the function $F : 2^{[T] \times U} \to [0, 1]$, $F(A) = \frac{1}{T}\sum_{t=1}^T f(A_t)$, where $A_t = \{u \in U : (t, u) \in A\}$ as:

$$\frac{r_i}{T} = \max_{a \in U} F(\widetilde{S}^{i-1} \cup \{a\}) - F(\widetilde{S}^i)$$

where $\widetilde{S}^\ell = \bigcup_{t=1}^T \{t\} \times S^\ell$. We show that $F(\widetilde{S}^i) - F(\widetilde{S}^{i-1}) \geq \frac{F(\widetilde{OPT}) - F(\widetilde{S}^{i-1})}{k} - \frac{r_i}{T}$, where $\widetilde{OPT}$ is the extension of $OPT = \text{argmax}_{|S|\le k} \sum_{t=1}^T f_t(S)$ to $[T] \times U$. Upon unrolling this recursion, we obtain the result.

To finish the proof of Theorem 6 we need to bound the overall regret of all $\mathcal{E}_i$. Observe that once we have fixed $S_1^{i-1}, \dots, S_T^{i-1}$, the feedback of expert i is completely determined since the elements $a_t^1, \dots, a_t^{i-1}$ depend only on experts $1, \dots, i-1$. Therefore, we have

$$\mathbb{E}[r_i \mid S_1^{i-1}, \dots, S_T^{i-1}] \le \eta T + \frac{\log |U|}{\eta}$$

by the Hedge regret guarantee. Integrating from k to 1 we get $\mathbb{E}[\mathcal{R}_T] \le \sum_{i=1}^k \mathbb{E}[r_i] \le k\left(\eta T + \frac{\log |U|}{\eta}\right)$, and the result follows with our choice of $\eta = \frac{\varepsilon}{k\sqrt{32T\log(k/\delta)}}$.

□

4 BANDIT SETTING

In the bandit case, the algorithm only receives as feedback the value $f_t(S_t)$. Given this restricted information, the algorithm must trade-off exploration of the function with exploiting current knowledge. As in (Streeter and Golovin, 2009), our algorithm controls this tradeoff using a parameter $\gamma \in [0,1]$, and by randomly exploring in each time-step independently with probability γ.

The non-private approach of Streeter and Golovin (2009) obtains $\mathcal{O}(T^{2/3})$ expected $(1-1/e)$-regret, and works as follows: In exploit rounds (prob. $1-\gamma$), play the experts' sampled choice S_t and feed back 0 to each $\mathcal{E}_i$. In explore rounds (prob. γ), select $i \in [k]$ and $a \in U$ uniformly at random. Play set $S_t = S_t^{i-1} + a$, observe feedback $f_t(S_t^{i-1} + a)$, give this value to $\mathcal{E}_i$, and feedback 0 to the remaining experts.

As we show in Appendix B.1, directly privatizing this algorithm using the Hedge method from the full-information setting results in an expected $(1-1/e)$-regret of $\mathcal{O}(T^{3/4})$, far from the known $\mathcal{O}(T^{2/3})$. The problem with this naive approach is that a new sample is obtained via the Hedge algorithms at every time-step, including exploit steps, so to ensure (ε,δ)-DP, a learning rate of $\eta = \frac{\varepsilon}{k\sqrt{32T\log(k/\delta)}}$ is required.

We improve upon this by calling the Hedge algorithm only after an exploration time-step has occurred, and new information is available. The learner continues playing this same set until the next exploration round, and privacy of these exploitation rounds follows from post-processing. This dramatically reduces the number of rounds that access the dataset, and reduces the overall amount of noise required for privacy.

If the exact number of exploration rounds were known, this could be plugged into the learning rate η to achieve (ε,δ)-DP. In the non-private setting, a *doubling trick* (see, e.g., Shalev-Shwartz et al. (2012)) can be employed to find the right learning rate by calling the algorithm multiple times, doubling T and thus doubling η on each iteration. Unfortunately, this doubling trick does not work in the private setting due to the direct non-linear connection between ε the privacy parameter, T the time horizon and η the learning rate, as specified in Proposition 2. Instead we use concentration inequalities (Alon and Spencer, 2004) to ensure that there are no more than $2\gamma T$ exploration rounds, except with probability $e^{-8T^{1/3}}$. With this, we can select a fixed learning rate $\eta = \frac{\varepsilon}{k\sqrt{32(2\gamma T)\log(k/\delta)}}$ and guarantee $\mathcal{O}(T^{2/3})$ expected $(1-1/e)$-regret, and the cost of $(\varepsilon, \delta + e^{-8T^{1/3}})$-DP.

One may wish to avoid the additional loss in the δ term. One possible approach is to try to trade off this loss with the regret guarantee. For instance, consider following the strategy from the previous paragraph as long as the number of explore times is at most $M = 2\gamma T$; if this number is exceeded, stop and guarantee nothing. This would ensure (ε,δ)-differential privacy by design. However, this method is also less likely to explore later time steps—e.g., in the extreme case $M = 1$, exploring later time steps is exponentially less likely than exploring earlier ones. In our regret analysis, uniformity over explore time steps is essential.

A fruitful way to avoid this δ term is by trading it off with space. In Appendix B.2 we show that this additional loss can be avoided by pre-sampling the exploration round. This requires $\Theta(T^{2/3} + k|U|)$ space, which may be unacceptable for large T.

Algorithm 3 presents the space-efficient approach. Here $\widehat{f}_t^i$ is the vector with a-th coordinate given by: $\widehat{f}_t^{i,a} = f_t(S_t^{i-1} + a)\mathbf{1}_{\{\text{Explore at time } t, \text{ pick } i, \text{ pick } a\}}$.

Theorem 7. *Algorithm 3 is $(\varepsilon, \delta + e^{-8T^{1/3}})$-DP.*

Theorem 8. *Algorithm 3 has $(1-1/e)$-regret*

$$\mathbb{E}[\mathcal{R}_T] \le \mathcal{O}\left(\frac{\sqrt{\log k/\delta}}{\varepsilon}(k(|U|\log|U|)^{1/3})^2 T^{2/3}\right).$$

Proof of Theorem 7 Observe that the algorithm only releases new information right an exploration time-step. If $t_1, \dots, t_M$ are the exploration time-steps, with M distributed as the sum of T independent Bernoulli random variables with parameter γ, then conditioned on the event $M < 2\gamma T$, we know that the outputs $S_1, S_{t_1+1}, \dots, S_{t_M+1}$ are (ε,δ)-DP by Theorem 5. Now, conditioning again on the event $M < 2\gamma T$, the entire output $(S_1, \dots, S_T)$ is (ε,δ)-DP since this corresponds to post-processing over the previous output by extending the sets to exploitation time-steps. We know that $M \ge 2\gamma T$ occurs w.p.

Algorithm 3: BanditDP(F, ε, δ)

Initialize: Set $\gamma = k\left(\frac{(16|U|\log|U|)^2}{T}\right)^{1/3}$ and $\eta = \frac{\varepsilon}{k\sqrt{32(2\gamma T)\log(k/\delta)}}$.
Instantiate k parallel copies $\mathcal{E}_1, \dots, \mathcal{E}_k$ of Hedge algorithm with rate η. Utilize each $\mathcal{E}_i$ to sample a_1^i and set $S_1 = \{a_1^1, \dots, a_1^k\}$.
for $t = 1, \dots, T$ **do**
  Sample $b_t \sim \text{Bernoulli}(\gamma)$.
  if $b_t = 1$ **then**
    Sample $i \in [k]$ u.a.r. and $a \in U$ u.a.r.
    Play $S_t^{i-1} \cup \{a\}$.
    Obtain value $f_t(S_t)$.
    Feed back the function $\widehat{f}_t^i$ to expert $\mathcal{E}_i$, $\forall i$.
    Utilize $\mathcal{E}_i$ to pick a_{t+1}^i $\forall i$.
    Update set $S_{t+1} = \cup_{i=1}^k \{a_{t+1}^i\}$.
  else
    Play S_t.
    Obtain $f_t(S_t)$.
    Update $S_{t+1} = S_t$.

$\leq e^{-8\gamma^2 T}$. Thus, for any S we have

$$
\begin{aligned}
&\Pr((\mathcal{E}_k, \dots, \mathcal{E}_1)(F) \in S) \\
&\leq \Pr((\mathcal{E}_k, \dots, \mathcal{E}_1)(F) \in S \mid M < 2\gamma T) \Pr(M < 2\gamma T) \\
&\quad + e^{-8\gamma^2 T} \\
&\leq e^{\varepsilon} \Pr((\mathcal{E}_k, \dots, \mathcal{E}_1)(F') \in S) + \delta + e^{-8\gamma^2 T}.
\end{aligned}
$$

The result now follows by plugging in the value of γ used in Algorithm 3. □

Proof of Theorem 8 Theorem 8 requires the following two lemmas, proved respectively in Appendices A.3 and A.4. The first lemma says that the $(1-1/e)$-regret experienced by the learner is bounded by the regret experienced by the expert and an additional error introduced during the exploration times. The second lemma bounds the regret experienced by the experts under the biased estimator.

Lemma 2. *If r_i denotes the regret experience by expert $\mathcal{E}_i$ in Algorithm 3, then*

$$
\left(1 - \frac{1}{e}\right) \max_{|S| \leq k} \sum_{t=1}^{T} f_t(S) - \mathbb{E}\left[\sum_{t=1}^{T} f_t(S_t)\right] \leq \sum_{i=1}^{k} \mathbb{E}[r_i] + \gamma T.
$$

Lemma 3. *If each $\mathcal{E}_i$ is a Hedge algorithm with learning rate $\eta = \frac{\varepsilon}{k\sqrt{32(2\gamma T)\log(k/\delta)}}$ then $\mathbb{E}[r_i] \leq 16\frac{k^2|U|\log|U|\sqrt{T\log(k/\delta)}}{\varepsilon\sqrt{\gamma}} + \frac{k|U|}{\gamma} T \cdot e^{-8\gamma^2 T}$.*

Using these two results with $\gamma = k\left(\frac{(16|U|\log|U|)^2}{T}\right)^{1/3}$:

$$
\begin{aligned}
&\mathbb{E}[\mathcal{R}_T] \\
&\leq k\left(16\frac{k^2|U|\log|U|\sqrt{T\log(k/\delta)}}{\varepsilon\sqrt{\gamma}}\right) + \frac{k|U|}{\gamma} T \cdot e^{-8\gamma^2 T} + \gamma T \\
&= \left(16\frac{k^3|U|\log|U|\sqrt{\log k/\delta}}{\varepsilon}\sqrt{\frac{T}{\gamma}} + \gamma T\right) + \frac{k|U|}{\gamma} T \cdot e^{-8\gamma^2 T} \\
&\leq 32\frac{\sqrt{\log k/\delta}}{\varepsilon}(k(|U|\log|U|)^{1/3})^2 T^{2/3} \\
&\quad + \frac{|U|^{1/3} T^{4/3}}{(16\log|U|)^{2/3}} e^{-8k^2(16|U|\log|U|)^{4/3} T^{1/3}}.
\end{aligned}
$$

□

5 EXTENSION TO CONTINUOUS FUNCTIONS

We sketch an extension of our methodology for (continuous) DR-submodular functions (Hassani et al., 2017; Niazadeh et al., 2018). Further details can be found in Appendix C.

Let $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$, where each $\mathcal{X}_i$ is a closed convex set in $\mathbb{R}$. A function $f : \mathcal{X} \to \mathbb{R}_+$ is called *DR-submodular* if f is differentiable and $\nabla f(\mathbf{x}) \geq \nabla f(\mathbf{y})$ for all $\mathbf{x} \leq \mathbf{y}$. DR-submodular functions are neither convex nor concave; however, they are concave in positive directions, which allows efficient approximation maximization. For instance, the multilinear extension of a submodular function (Calinescu et al., 2011) is DR-submodular. The function f is said to be β-smooth if $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq \beta\|\mathbf{x} - \mathbf{y}\|_2$, for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. In the online learning DR-submodular maximization problem, at each time-step $t = 1, \dots, T$, a β-smooth DR-submodular function $f_t : \mathcal{X} \to [0, 1]$ arrives and, without observing the function, the learner selects a point $\mathbf{x}_t \in \mathcal{X}$ learned using $f_1, \dots, f_{t-1}$. She gets the value $f_t(\mathbf{x}_t)$ and also oracle access to ∇f_t. The learner's goal is to minimize the $(1-1/e)$-regret

$$
\mathcal{R}_T = \left(1 - \frac{1}{e}\right) \max_{\mathbf{x} \in \mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{x}) - \sum_{t=1}^{t} f_t(\mathbf{x}_t).
$$

Online DR-submodular problems have been extensively studied in the full information setting—see for instance (Chen et al., 2018b,a; Niazadeh et al., 2018). Similarly to the discrete submodular case, most of these methods implement K ordered algorithms $\mathcal{E}_0, \dots, \mathcal{E}_{K-1}$ for optimizing linear functions over $\mathcal{X}$. Algorithm $\mathcal{E}_k$ computes a direction of maximum increment from a point given by the algorithms $\mathcal{E}_{k-1}, \dots, \mathcal{E}_0$. The learner averages these directions to obtain a new point to play in the region $\mathcal{X}$. This is the continuous version of the Hedge approach.

We show in Algorithm 4 and Theorem 9 that a simple modification transforms the continuous method of Chen et al. (2018b) into a differentially private one. For this, we utilize the Private Follow the Approximate Leader (PFTAL) framework of Thakurta and Smith (2013) as a black-box. PFTAL is an online convex optimization algorithm for minimizing L-Lipschitz convex functions over a compact convex region $\mathcal{X}$. In few words, their algorithm guarantees $(\varepsilon, 0)$-DP and achieves an expected regret $\mathcal{O}\left(\frac{L^2\sqrt{nT\log^{2.5}T}}{\varepsilon}\right)$.

Algorithm 4: $(F = \{f_t\}_{t=1}^T, \varepsilon)$

Let $K = \left(\frac{T}{\log^{2.5}T}\right)^{1/4}$. Initialize $\mathcal{E}_0, \ldots, \mathcal{E}_{K-1}$ parallel copies of PFTALs with privacy parameter $\varepsilon' = \varepsilon/K$.
for $t = 1, \ldots, T$ **do**
  for $k = 0, \ldots, K-1$ **do**
    Let $\mathbf{v}_t^k$ be vector found using $\mathcal{E}_k$.
  Let $\mathbf{x}_t = \frac{1}{K}\sum_{k=0}^{K-1}\mathbf{v}_t^k$.
  Play $\mathbf{x}_t$, receive $f_t(\mathbf{x}_t)$ and access to ∇f_t.
  Feed back each $\mathcal{E}_k$ with the linear obsective $\ell_k(\mathbf{v}) = \nabla f_t(\mathbf{x}_t^k)^\top \mathbf{v}$ where $\mathbf{x}_t^k = \frac{1}{K}\sum_{i=0}^{k-1}\mathbf{v}_t^i$.

Theorem 9 (Informal)**.** *Algorithm 4 is* $(\varepsilon, 0)$*-DP with expected* $(1 - 1/e)$*-regret*

$$\mathcal{O}\left(\frac{T^{3/4}(\log^{2.5}T)^{1/4}}{\varepsilon}\right).$$

The big $\mathcal{O}$ term hides dimension, bounds in gradient and diameter of $\mathcal{X}$ and only shows terms in T and privacy parameter ε. The proof appears in Appendix C.

Acknowledgments

We thank all the anonymous reviewers for their valuable comments. R.C. was supported in part by NSF grants CNS-1850187 and CNS-1942772 (CAREER), a Mozilla Research Grant, and a JPMorgan Chase Faculty Research Award. This work was completed while R.C. was at Georgia Institute of Technology.

References

Ahmed, A., Teo, C. H., Vishwanathan, S., and Smola, A. (2012). Fair and balanced: Learning to present news stories. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 333–342. ACM.

Alon, N. and Spencer, J. H. (2004). *The probabilistic method*. John Wiley & Sons.

Bach, F. et al. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373.

Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., and Krause, A. (2014). Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680. ACM.

Bian, A. A., Mirzasoleiman, B., Buhmann, J., and Krause, A. (2017). Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766.

Cardoso, A. R. and Cummings, R. (2019). Differentially private online submodular minimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1650–1658.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Chatterjee, P., Hoffman, D. L., and Novak, T. P. (2003). Modeling the clickstream: Implications for web-based advertising efforts. *Marketing Science*, 22(4):520–541.

Chen, L., Harshaw, C., Hassani, H., and Karbasi, A. (2018a). Projection-free online optimization with stochastic gradient: From convexity to submodularity. *arXiv preprint arXiv:1802.08183*.

Chen, L., Hassani, H., and Karbasi, A. (2018b). Online continuous submodular maximization. *arXiv preprint arXiv:1802.06052*.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. (2010a). Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010b). Boosting and differential privacy. In *2010 IEEE*

51st Annual Symposium on Foundations of Computer Science, pages 51–60. IEEE.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652.

Fisher, M. L., Nemhauser, G. L., and Wolsey, L. A. (1978). An analysis of approximations for maximizing submodular set functions—ii. In *Polyhedral combinatorics*, pages 73–87. Springer.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. (2010). Differentially private combinatorial optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1106–1125. Society for Industrial and Applied Mathematics.

Hassani, H., Soltanolkotabi, M., and Karbasi, A. (2017). Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pages 5841–5851.

Hazan, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325.

Jain, P., Kothari, P., and Thakurta, A. (2012). Differentially private online learning. In *Conference on Learning Theory*, pages 24–1.

Krause, A. and Golovin, D. (2014). Submodular function maximization.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–103.

Mirrokni, V., Schapira, M., and Vondrák, J. (2008). Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77. ACM.

Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. (2017). Differentially private submodular maximization: data summarization in disguise. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2478–2487. JMLR. org.

Nemhauser, G. L. and Wolsey, L. A. (1978). Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188.

Niazadeh, R., Roughgarden, T., and Wang, J. (2018). Optimal algorithms for continuous non-monotone submodular and dr-submodular maximization. In *Advances in Neural Information Processing Systems*, pages 9594–9604.

Rafiey, A. and Yoshida, Y. (2020). Fast and private submodular and k-submodular functions maximization with matroid constraints. *arXiv preprint arXiv:2006.15744*.

Schrijver, A. (2003). *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media.

Shalev-Shwartz, S. et al. (2012). Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194.

Streeter, M. and Golovin, D. (2009). An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584.

Streeter, M., Golovin, D., and Krause, A. (2009). Online learning of assignments. In *Advances in neural information processing systems*, pages 1794–1802.

Tang, L., Jiang, Y., Li, L., and Li, T. (2014). Ensemble contextual bandits for personalized recommendation. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 73–80.

Thakurta, A. G. and Smith, A. (2013). (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pages 2733–2741.

Williamson, D. P. and Shmoys, D. B. (2011). *The design of approximation algorithms*. Cambridge university press.

Zhang, B., Wang, N., and Jin, H. (2014). Privacy concerns in online recommender systems: Influences of control and user data input. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*, pages 159–173.

Zhang, M., Shen, Z., Mokhtari, A., Hassani, H., and Karbasi, A. (2020). One sample stochastic frank-wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936.